Probability for Data Science – Master's Degree in Data Science – A.Y. 2023/2024
Academic Staff: Francesca Collet, Paolo Dai Pra

PROJECT 1: COVER TIME OF SIMPLE RANDOM WALKS ON A GRAPH

Abstract. A simple random walk on a graph is a sequence of movements from one vertex to another where at each step the next vertex is chosen uniformly at random from the neighbors of the current one. Central to this project is the cover time of a walk on a graph, that is, the expectation of the number of steps required to visit every vertex, maximised over all the starting vertices. The aim is to investigate how running many random walks in parallel yields a speed-up in the cover time.

Random walk on graphs

An **undirected graph** $G = (V, E)$ consists of a *vertex set* V and an *edge set* E, where the elements of E are unordered pairs of vertices: $E \subseteq \{\{i, j\} : i, j \in V,\, i \neq j\}$. We can think of V as a set of dots, where two dots i and j are joined by a line if and only if $\{i, j\}$ is an element of the edge set. When $\{i, j\} \in E$, we write $i \sim j$ and say that j is a *neighbor* of i (and also that i is a neighbor of j). The *degree* $\deg(i)$ of a vertex $i \in V$ is the number of its neighbors.

Example (Complete graph)*.* The complete graph is a graph in which every pair of distinct vertices is connected by a unique edge.

Given a graph $G = (V, E)$, we can define **simple random walk on** G to be the Markov chain with state space V and transition matrix $\mathbb{P} = (P_{ij})_{i,j \in V}$, where

$$P_{ij} = \begin{cases} \dfrac{1}{\deg(i)} & \text{if } j \sim i \\ 0 & \text{otherwise.} \end{cases}$$

That is to say, when the chain is at vertex i, it examines all the neighbors of i, picks one uniformly at random, and moves to the chosen vertex.

Algorithm to simulate a random walk on a graph

Choose your favorite graph $G = (V, E)$, with $V = \{1, 2, \dots, n\}$. As initial distribution for the Markov chain take the simplest possible; assume it is concentrated in a single vertex $v \in V$. The following algorithm produces the first N steps $X_1, \dots, X_N$ taken by the random walk on the graph G.

```
Input:   graph $G = (V, E)$;
         initial vertex $v \in V$;
         number of steps $N$;

Output:  trajectory of the simple random walk starting at $v$;

Procedure

Step 1. Set $X_0 = v$ and $t = 1$.

Step 2. When $X_t = i$,

    1. choose a vertex $j$ uniformly at random from the set of neighbors of vertex $i$;

    2. set $X_{t+1} = j$.

Step 3. If $t < N$ set $t = t + 1$ and return to Step 2, else stop.
```

Cover time of a graph

Let $G = (V, E)$ be a finite graph and let $X = (X_t)_{t\in\mathbb{N}}$ be a random walk on G. The cover time $T_{\text{cov}}(G)$ of the simple random walk X on G is the expected number of steps needed for the chain to visit all the vertices in V, maximized over all the initial positions. More precisely, if

$$T_v(G) = \min\big\{t \geq 1 : \{v, X_1, \ldots, X_t\} = V\big\}$$

denotes the first time at which all the nodes have been visited by the random walk started at v, then the cover time is defined as

$$T_{\text{cov}}(G) = \max_{v\in V} E(T_v).$$

We turn now our attention to the case of parallel random walks evolving on G. Let $X^1 = (X_t^1)_{t\in\mathbb{N}}$, $X^2 = (X_t^2)_{t\in\mathbb{N}}, \ldots, X^\kappa = (X_t^\kappa)_{t\in\mathbb{N}}$ be κ independent random walks on G, all starting from the same vertex $v \in V$. The natural extension of the definition of cover time is the κ-cover time $T_{\text{cov}}^\kappa(G)$, that is the expected number of steps for each node in V to be visited by at least one of the walks, maximized over all the initial nodes. Formally, if

$$T_v^\kappa(G) = \min\{t \geq 1 : \{v, X_1^1, \ldots, X_t^1, X_1^2, \ldots, X_t^2, \ldots, X_1^\kappa, \ldots, X_t^\kappa\} = V\}$$

represents the time taken by κ simple random walks, all starting at v, to visit all nodes in G, then the κ-cover time is defined as

$$T_{\text{cov}}^\kappa(G) = \max_{v\in V} E(T_v^\kappa).$$

Remark. The expected time $E(T_v)$ (resp. $E(T_v^\kappa)$) can be estimated by exploiting the law of large numbers. Let M denote the number of independent realizations of a simple random walk on G (resp. of an independent family of κ random walks on G), started at the vertex v. Moreover, let $T_v[j]$ (resp. $T_v^\kappa[j]$) be the number of steps taken by the j-th simulation of the walk (resp. of the family of κ walks) started at v to cover the whole graph. If M is *sufficiently large*, then we have the approximations

$$E(T_v) \approx \frac{1}{M}\sum_{j=1}^{M} T_v[j] \quad \text{and} \quad E(T_v^\kappa) \approx \frac{1}{M}\sum_{j=1}^{M} T_v^\kappa[j].$$

Project

We want to show that if we think of κ as a function of the graph size and choose it correctly, then κ random walks can cover a complete graph κ times faster than a single random walk.
Consider a *complete graph* G on n vertices. By running several simulations and collecting the results in appropriate plots, show that

- the cover time $T_{\text{cov}}(G)$ grows as $cn\log n$, for some constant $c > 0$.
- if $\kappa = \kappa(n) = n$, the κ-cover time $T_{\text{cov}}^\kappa(G)$ grows as $c\log n$, for some constant $c > 0$.

Perform the analysis for graph sizes of the form $n = 500\alpha$, with $\alpha \in \{1, 2, \ldots, 20\}$ (integer numbers from 1 to 20).

Remark. Notice that, if G is the complete graph, then $T_{\text{cov}}(G) = E(T_v)$ and $T_{\text{cov}}^\kappa(G) = E(T_v^\kappa)$, as the collections $(T_v)_{v\in V}$ and $(T_v^\kappa)_{v\in V}$ are families of i.i.d. random variables.

References

[1] Alon N., Avin C., Koucký M., Kozma G., Lotker Z., and Tuttle M.R., Many random walks are faster than one, *Combinatorics, Probability and Computing*, 20:481–502, 2011

[2] Levin D.A. and Peres Y., *Markov chains and mixing times*, Volume 107, American Mathematical Society, 2017

[3] Ross S.M., *Simulation*, Academic Press, 2006